University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitní 8, CZ-30614 Plzeň
Czech Republic

LRecorder
LICS LASER Project Auxiliary Software
—
Documentation

Application Version 1.04-R08/04
Documentation Version 1.00

Written by Kamil Ekštein

1 Introduction

The LRecorder application is an auxiliary software which is a part of and helps in the development process of the **LICS**[1] **Automatic Speech Evaluation/Recognition** (hereinafter referred to as LASER) framework. LRecorder can be (and mostly is) used without the framework as a stand-alone application.

The purpose of the LRecorder application is to automatize and ease the phase of collecting a corpus material used to train an ASR system. Simply speaking, LRecorder is an audio recorder software which works according to a *scenario*.

2 System Requirements

The LRecorder application can be run on any Intel- or AMD-based PC workstation[2] under any Win32-based operating system like e.g. Microsoft Windows 95, 98, Me, NT, 2000, and XP. It was **not tested** under Microsoft Windows Vista, however, there is no objective reason for it not to run under Vista. The application can be run also in any virtual machine running one of the above mentioned operating systems. The application runs in Wine environment under Linux too (however only a limited testing of basic functionality was performed).

The application is undemanding — it is based on pre-DirectX low level Win32 Multimedia System calls and basic Windows GUI components. A minimum system configuration is:

- 200 MHz 32-bit (x86) Intel or AMD (or clone) processor
- 256 MB of system memory
- 1 MB of hard drive space for the application
- approx. 172 KB of hard drive space for every second of recorded signal
- CD-ROM drive
- Soundcard capable of 16-bit/44.1 kHz recording and playback

3 Installation

There is no specific installation procedure. Simply unzip the downloaded archive into a new empty folder. The folder should afterwards contain the following files:

1. `LRecorder.exe` — the application,

[1] **L**aboratory of **I**ntelligent **C**ommunication **S**ystems, Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia in Plzeň, Czech Republic. `http://liks.fav.zcu.cz`.

[2] or clones

2. `LRecorder.ini` — the configuration file with initial values of important user-adjustable variables,

3. `beep.pcm` — the beep sound indicating that the recording has started.

The application is started by launching the `LRecorder.exe` executable on Windows systems or by launching the Wine environment with the `LRecorder.exe` as its parameter on Linux systems.

4 LRecorder Description

After launching the application the main window will look like on the screenshot below (except for the number marks, of course):

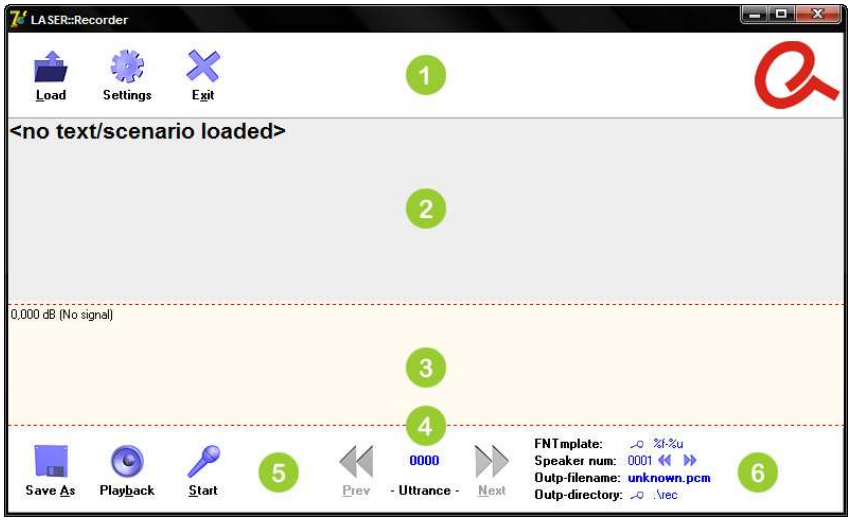

Fig. 1.: Initial window of the LRecorder application.

A detailed description of the numbered areas follows:

1. **Toolbar** — Enables the user to initiate basic functionality activities: Loading a text to read into a recording, setting user-adjustable options of the application, and exiting the application. To initiate the mentioned activities, left-click the appropriate button. A version and copyright information can be displayed by double-clicking the big Q letter in the upper right corner of the main window.

2. **Text/Scenario Display** — The application displays a loaded text or scenario sentence by sentence in this area. A user can browse through the text/scenario either by pressing <**Page Up**> and <**Page Down**> keys or by left-clicking the double arrow buttons in the area 4.

3. **Signal Display** — A waveform of a recorded utterance is displayed in this area right after the recording was finished. It enables the user to perform a basic visual inspection whether the signal is all right. Moreover the application displays (in the upper left corner of the area) textual information about the signal quality: If the signal is acceptable to be stored

in an audio corpus, an average volume of the signal is displayed in decibels in black. In the case the signal is too weak to act as a training material for an ASR system, the average volume is displayed in **bold red** together with the word "Low" indicating the signal volume is too weak. On the other hand if the signal volume is too high causing the signal to be clipped, the expression "Wave malformation (Clip)" is displayed under the volume level. **Either if the signal is too weak or clipped it is *strongly advised* not to include it into a training corpus.**

4. **Position Panel** — The two double arrow buttons enable the user to browse through the text/scenario forwards (by pressing <**Page Down**> too) or backwards (by pressing <**Page Up**> too). Right in between these buttons there is an indicator (number) of the active utterance (sentence or scenario item) to be recorded. If the buttons are disabled (gray) the text/scenario has not been yet loaded and thus it is not possible to browse through it.

5. **Control Panel** — The control panel enables the user to (from left to right): (i) **save** the currently recorded signal to a file system under a different name from the preselected naming scheme (see later) or into a different location; (ii) **play back** the currently recorded signal (also by pressing <**Backspace**>); (iii) **start/stop the recording** of the current utterance (also by pressing <**Enter**>).

6. **Storage Settings Panel** — This panel enables the user to adjust the options that control how the application stores the recorded utterances to the file system. For detailed description see section 5-2 Automatic Storage.

5 Working with LRecorder

5-1 Preparing texts/scenarios

The input texts or scenarios are ordinary plain ASCII text files structured into sentences terminated by the CRLF[3] character pairs. Encoding of national alphabet characters is not regarded in the application: The LRecorder application simply displays verbatim the contents of an input file using the code page settings of the underlying operating system.

The contents of the text input file are typically CRLF-terminated sentences (character strings) that the user should read for the recording. They are displayed separately, one by one. To move to the next sentence during the recording stage, the <**Page Down**> key should be pressed. To move to the previous one, the <**Page Up**> should be pressed.

An input scenario is virtually the same, however, the text to read is usually enriched with a certain information about how to read it or simply an instruction what the user should say for the recording (like e.g. "Please, state your name now.").

[3] CR = Carriage Return, LF = Line Feed.

5-2 Automatic Storage

Each recorded signal is automatically saved to the file system. Right after the recording is finished the application saves the signal as **raw signed 16-bit integer monaural PCM data**[4].

The location and name of the saved file is determined by the settings in the **Storage Settings Panel** (5). The **FNTmplate** (File Name Template) specifies how the file name is composed from known facts about the utterance and input text/scenario. The user can change the value by pressing the "Magnifying glass" button next to the template entry:

Variable	Meaning
`%f`	the file name of the loaded input text/scenario
`%s`	the number (order) of the speaker
`%u`	the number (order) of the utterance from the beginning of the input text/scenario

When creating a file name for the recording the above listed variables (starting with `%` sign) are translated into their respective meanings. Any other characters in the template are left intact and copied into the file name without any change. It naturally implies that the template cannot contain any characters with a special meaning in the file system like e.g. `*`, `?`, \, etc.

The **Speaker num** (Speaker number) indicates order of the speaking (recorded) user. The actual number can be changed by pressing the small double arrow buttons next to the label. If the file name template contains the `%s` variable, this number is projected into the names of the saved recording files. Otherwise it is ignored.

The **Outp-filename** (Output file name) label indicates the name of the file the current recording is (will be) stored in. The actual file name is composed (via the above depicted process) from either variables or constant characters stated in the file name template field.

The **Outp-directory** (Output directory[5]) determines in which directory the recording files are (will be) stored. The value can be changed by pressing the "Magnifying glass" button next to the entry.

5-3 Recording the utterances

After loading an input text or scenario file the first utterance (sentence, part of the scenario) is displayed as shown on the figure 2 below.

[4]Raw means there is no header in the resulting file. Only the signal samples are saved — it is upon the user/corpus administrator to remember and maintain the right recording settings like e.g. the sampling rate. PCM stands for Pulse Code Modulation technique of digitising an analogue signal. To get more details on the topic see literature.

[5]Within the context of certain operating systems the word "folder" is used instead of "directory". Meaning is, however, the same.

Fig. 2.: The LRecorder application window with a scenario loaded.

When the speaker is ready to utter for the recording he or she (or another user acting as an operator) presses the <**Enter**> key and the application produces a beep and starts to record the audio signal. The speaker should start speaking after the beep. Right after the utterance is over the operator presses the <**Enter**> key once more and the recording is stopped and the data are written down to an appropriate file. **Pressing any other keys except for <Enter> during the recording may lead to unpredictable results**.

As the utterance is recorded the application shows the waveform together with state information in the upper left corner of the signal display. If an erroneous state is indicated (i.e. Low or Clip, see above) the utterance should be recorded once more, preferably under slightly changed conditions (a microphone positioned nearer or further from the speaker's mouth).

The figures 3 a 4 below depict two most typical errors that can appear during the recording: The first one (fig. 3) is marked as **Wave malformation** and happens when the input signal was so loud that it did not fit into the total dynamic range of the A/D converter (soundcard). To get rid of this problem it is necessary either (i) to position the microphone further from the speaker's mouth (preferred), or (ii) to speak more quietly (not preferred as it is harder to maintain the same diminished voice level). If the soundcard or preamplifier of the microphone has a gain selector the problem can be also solved by decreasing the gain level.

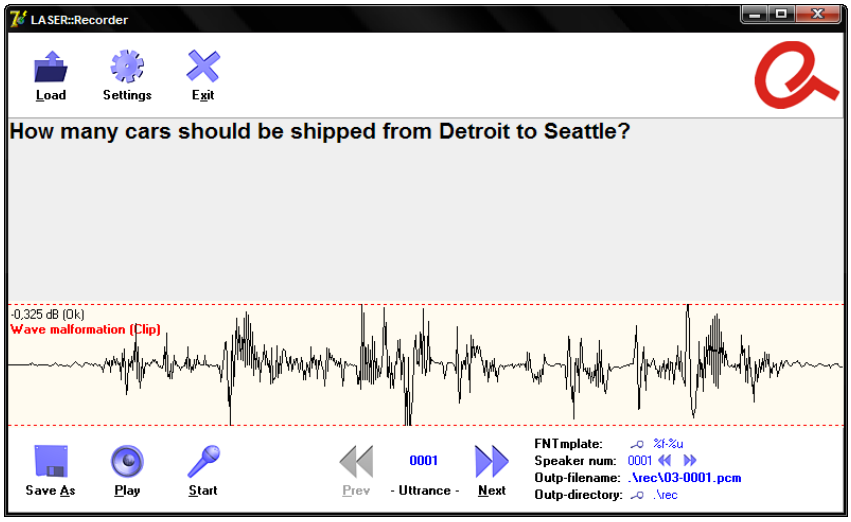

Fig. 3.: The recorded signal is clipped.

The second error (fig. 4) is when the signal is too weak. Again it can be solved by moving the microphone closer to the speaker's mouth or speaking more loudly or increasing the gain (if possible).

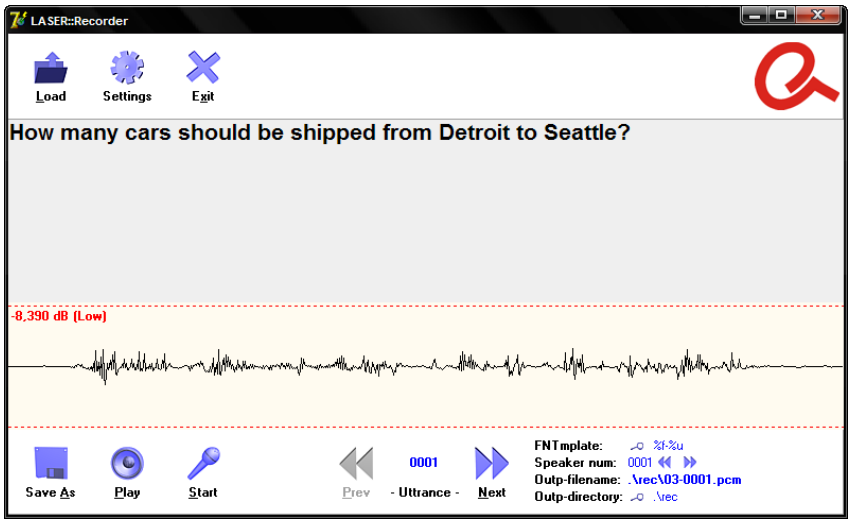

Fig. 4.: The LRecorder application window with a scenario loaded.

If a file already exists in the file systems (which happens when recording the same sentence once more) the application asks whether the new recording should overwrite the old one.

After completing the recording of a particular sentence/scenario entry the user presses <**Page Down**> to move to another one. If there are no more sentences or items in the scenario the application displays `<End-Of-File>` in red in the text display area.

6 LRecorder Application

The LRecorder application was used (and originally developed) for the **LICS Audio Corpus** collection. In total 12:43 hours of speech uttered by more than 100 speakers was recorded with the LRecorder. The mentioned set of partial (differently focused) corpora is used to train the **LICS LASER** speech recognizer.

It was also used to create templates for a concatenation synthesizer built at LICS. In both these tasks the LRecorder proved to be a robust, stable and useful tool.

7 LRecorder Licensing Policy

The LRecorder application is published under the **Creative Commons Attribution-No Derivative Works 3.0 Unported License**. A human-readable summary of the Legal Code (the full license) is available at the Creative Commons' web page at `http://creativecommons.org/licenses/by-nd/3.0/`.